Stefan Fischerländer

Ein Vektor sagt mehr als tausend Wörter

AlphaGo, RankBrain und Google AI – Machine-Learning und künstliche Intelligenz sind für Suchmaschinenoptimierer inzwischen geflügelte Worte. Was allerdings Machine-Learning ist und wie Suchmaschinen es einsetzen, ist wenig geläufig. Grund genug für Stefan Fischerländer, die Grundlagen der Methoden vorzustellen und aufzuzeigen, was mit von Google veröffentlichter Software heute bereits möglich ist. Die Ergebnisse sind beeindruckend, öffnen die Augen und geben einen zumindest groben Eindruck, wie weit eine inhaltliche Beurteilung aufgrund von Texten bzw. deren Worten durch Maschinen gehen kann.

Es vergeht derzeit kaum eine SEO-Konferenz ohne einen Vortrag über die Bedeutung von Machine-Learning für Google. Und auch Sundar Pichai, Googles CEO, erklärte dazu bereits im Oktober 2015: „Wir werden Machine-Learning in all unseren Produkten einführen, in der Suche, in der Werbung, bei Youtube oder im Play Store." Doch auch wenn Pichai Suche an erster Stelle seiner Auflistung nannte, so gehen die Beispiele in den Vorträgen oder Blogposts doch meist auf andere, eher visuelle oder spielerische Methoden ein. Allerdings wurde zuletzt immer deutlicher, dass spätestens mit dem relativ neuen, als RankBrain bezeichneten Teil des Google-Algorithmus Machine-Learning-Methoden auch in der normalen Websuche Einzug hielten. Da RankBrain nach Googles eigener Darstellung der drittwichtigste Rankingfaktor ist, sollte dies für jeden Online-Marketer Anlass genug sein, einen genaueren Blick auf Machine-Learning zu werfen.

Einer Definition von Arthur Samuel, dem Erfinder des Begriffs, zufolge, bezeichnet Machine-Learning Methoden, mit denen Computer lernen können, Aufgaben zu lösen, ohne dass sie explizit zur Lösung dieser Aufgabe programmiert wurden. Eine für Suchmaschinen nicht ungewöhnliche Problemstellung, die in diese Kategorie fällt, ist die Klassifizierung. Ein Algorithmus versucht dabei, Objekte möglichst passend in vorgegebene Gruppen (Klassen) einzusortieren. Google zeigt bei Suchanfragen mit Kaufabsicht (transactional queries) beispielsweise häufig ausschließlich Online-Shops und Preissuchmaschinen. Also hat die Suchmaschine irgendwo ein Verfahren im Einsatz, das Websites in Klassen wie „Online-Shop" oder „Preissuchmaschine" einordnen kann.

Objekte klassifizieren

Wie Computer diese Aufgabe angehen, soll eine Methode zeigen, die Automodelle klassifizieren kann. Autos lassen sich durch verschiedenste Größen charakterisieren; zwei besonders aussagekräftige Angaben sind die Leistung in Kilowatt und das Leergewicht in Kilogramm. Für sechs Modelle sind diese Angaben in der Tabelle 1 dargestellt.

Allgemein formuliert, enthält die Tabelle sechs Objekte (Automodelle) mit jeweils zwei Variablen (Leistung und Gewicht). Somit lassen sich die Objekte in ein Koordinatensystem einzeichnen, das in der x-Richtung die Leistung darstellt und in der y-Richtung das Gewicht. Für die Daten aus Tabelle 1 ergeben sich somit die blauen Punkte in Abbildung 1.

	Leistung/kW	Gewicht/kg
Opel Corsa	51	1120
BMW X6	225	2100
Mercedes B-Klasse	90	1395
Ford Transit	74	2043
Mercedes Sprinter	65	2030
Ford Expedition	223	2626

Tabelle 1

DER AUTOR

Stefan Fischerländer ist Geschäftsführer der Passauer Digitalagentur Gipfelstolz. Er arbeitet seit mehr als 15 Jahren als SEO-Berater und beschäftigt sich vor allem mit den strategischen Herausforderungen des digitalen Marketings.

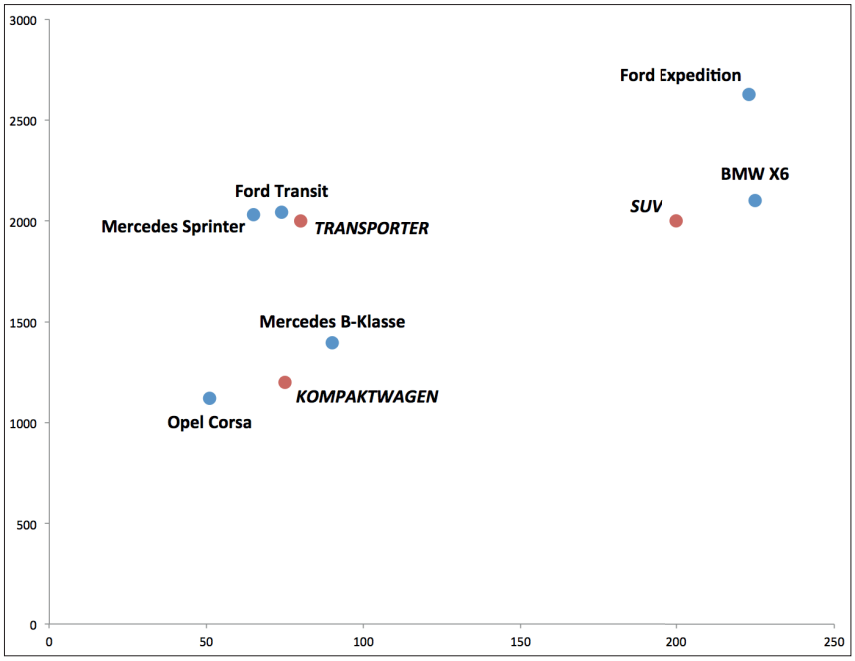

Abb. 1: Die verschiedenen Automodelle sind hier als blaue Punkte abhängig von ihrer Leistung (in kW) nach rechts und von ihrem Gewicht (in kg) nach oben aufgetragen; die roten Punkte stehen für die vorgegebenen Klassen

Ein Mensch kann aus dieser grafischen Darstellung bereits viele Informationen entnehmen. In der Grafik benachbarte Automodelle sind sich offenbar sehr ähnlich, während Autos mit großem Abstand – etwa Corsa und X6 – ziemlich unterschiedliche Modelle sind. Zudem erkennt ein Mensch sofort, dass hier offenbar drei Klassen an Autos dargestellt sind.

Geringer Abstand bedeutet große Ähnlichkeit

Mathematisch betrachtet sind die Automodelle jeweils zweidimensionale Vektoren. Der Opel Corsa etwa wird durch den Vektor (51 1120) dargestellt, der BMW X6 durch den Vektor (225 2100). Diese Darstellung beliebiger Objekte als Vektoren bildet die Basis fast aller Machine-Learning-Verfahren und die hier getroffene Annahme, dass zwei Vektoren mit geringem Abstand recht ähnlich sind, gilt für die meisten derartigen Methoden.

Die Abstände von Vektoren (hier als Ortsvektoren betrachtet, die im Ursprung des Koordinatensystems platziert werden) lassen sich leicht durch die euklidische Distanz berechnen. Für den zweidimensionalen Fall lautet die entsprechende Formel:

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2},$$

wobei (x1 x2) und (y1 y2) die beiden Ortsvektoren sind, deren Distanz zu berechnen ist.

Der Abstand von Corsa und X6 berechnet sich damit als:

$$d = \sqrt{(51 - 225)^2 + (1120 - 2100)^2} = 995{,}3$$

Für Corsa und B-Klasse ergibt sich hingegen lediglich ein Abstand von 277,8.

Zur Klassifizierung benötigt das Verfahren noch die vordefinierten Klassen, denen die Objekte (Automodelle) zugeordnet werden sollen. Solche Klassen lassen sich durch die Angabe typischer Größen definieren; die drei Klassen „Kompaktwagen“, „SUV“ und „Transporter“ könnten demnach beispielsweise durch die Werte in Tabelle 2 definiert werden. Diese Klassenrepräsentanten sind in Abbildung 1 als rote Punkte dargestellt.

Eine sehr einfache Methode, die sechs Automodelle den Klassen zuzuordnen, besteht nun darin, für jedes Auto den Abstand zu jeder der vorgegebenen Klassen zu berechnen und das Auto dann in die Klasse zu packen, für die der Abstand am kleinsten ist. Diese Methode wird als Abstandsklassifikator bezeichnet und ist wohl eines der einfachsten Machine-Learning-Verfahren überhaupt.

	Leistung/kW	Gewicht/kg
Kompaktwagen	75	1200
SUV	200	2000
Transporter	80	2000

Tabelle 2

Rechnen mit beliebig vielen Dimensionen

Das gewählte Beispiel mit lediglich zwei Variablen (Gewicht und Leistung) für die Autos ist natürlich sehr einfach und die allermeisten Objekte, mit denen Computer rechnen müssen, haben deutlich mehr solcher Variablen. Eine Aufgabenstellung mit mehr als zwei Dimensionen lässt sich zwar nicht mehr zeichnerisch darstellen, aber das eben gezeigte Verfahren benötigt diese zeichnerische Darstellung ja gar nicht. Es reicht, wenn der Abstand zwischen zwei Objekten (Vektoren) berechnet werden kann – und das geht mit der euklidischen Distanz für beliebig viele Dimensionen. Im allgemeinen Fall lautet die zuvor für zwei Dimensionen gezeigte Formel für den Abstand:

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2},$$

wobei n die Anzahl der Dimensionen ist.

Neben dem eben gezeigten Verfahren, das Objekte klassifizieren kann, gibt es noch viele weitere Verfahren: Cluster-Algorithmen etwa benötigen keine vorgegebenen Klassen, sondern versuchen selbstständig, Gruppen (Cluster) von ähnlichen Objekten zu bilden. Google News stellt beispielweise Artikel zum gleichen Thema zusammen

dar; dazu ist im Hintergrund ein Cluster-Algorithmus nötig, der erkennt, welche Artikel gleichartig sind und entsprechend dargestellt werden müssen. Verfahren zur Dimensionsreduzierung vereinfachen gegebene Daten, sodass die wesentlichen Eigenschaften besser zutage treten oder sich in einer Grafik darstellen lassen. Andere Verfahren wiederum sind im Einsatz, um Ausreißer in Daten zu entdecken oder Vorhersagen zu treffen, etwa die berühmte Kauften-auch-Funktion von Amazon. Obwohl die Ergebnisse dabei recht intelligent wirken können, haben die Verfahren mit künstlicher Intelligenz nicht viel zu tun. Es handelt sich vielmehr um statistische Methoden, deren Ergebnisse eine Wissensbasis bilden können, die dann eine Grundlage wirklicher künstlicher Intelligenz sein kann.

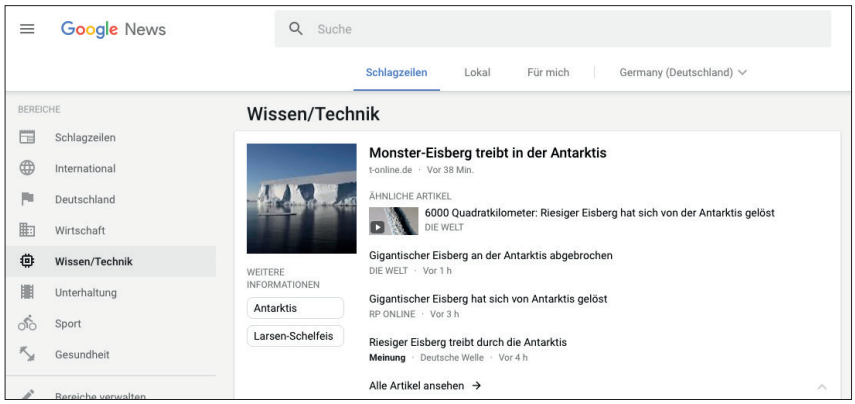

Abb. 2: Google News stellt Nachrichtenartikel zum gleichen Thema gebündelt dar; dahinter steckt mit Clustering eine klassische Machine-Learning-Methode

Dokument	Inhalt
1	Paris ist die Hauptstadt von Frankreich und die größte Stadt in Frankreich
2	Paris ist eine Stadt in Frankreich
3	Paris ist eine Metropole

Tabelle 3

Das Vector-Space-Model

So unterschiedlich der Einsatzzweck der verschiedenen Verfahren auch sein mag, im Kern besteht Machine-Learning stets aus zwei Schritten: Der erste Schritt ist, Objekte als Vektoren darzustellen. Das mag, wie im Autobeispiel vorhin, trivial sein. Für andere Aufgabenstellungen ist das aber häufig die eigentliche Schwierigkeit. Der zweite Schritt ist dann, mit den so gewonnenen Vektoren zu rechnen; etwa Abstände zu ermitteln oder Vektoren zu finden, die in eine ähnliche Richtung zeigen. Sobald ein Problem auf Vektoren zurückgeführt wurde, können alle Methoden darauf angewendet werden, ohne auf die konkrete Aufgabenstellung einzugehen. Für Suchmaschinen ist das deshalb so interessant, weil im klassischen Information Retrieval – der Grundlage textbasierter Suchmaschinen – Dokumente (Webseiten) als Vektoren dargestellt werden.

In diesem Vector-Space-Model sind die Objekte, die als Vektoren dargestellt werden, die zu indexierenden Webseiten. Doch wie wird aus einem Text ein Vektor? Dazu wird für jeden Text ein Vektor erstellt, dessen Dimension gleich der Anzahl aller Wörter ist, die wenigstens in einem der zu indexierenden Texte vorkommen. Jede Dimension steht für ein Wort; kommt ein Wort in einem Dokument vor, steht an dieser Stelle im Vektor eine Zahl ungleich null. Welche Zahl genau dort steht, hängt von der konkreten Implementierung ab und wird durch Formeln bestimmt, die für Suchmaschinenoptimierer bekannt klingen dürften: TF-IDF und WDF-IDF sind zwei recht häufig benutzte Berechnungsmethoden. Für beide Verfahren wird die Zahl größer, wenn ein Wort in dem Dokument häufiger vorkommt; sie wird hingegen kleiner, wenn ein Wort in sehr vielen verschiedenen Dokumenten auftaucht.

Dieses Verfahren lässt sich an einem konkreten Beispiel veranschaulichen. Für die in Tabelle 3 aufgeführten Dokumente sind Vektoren zu erstellen. Die unwichtigen Wörter wie ist oder die bleiben der Einfachheit halber dabei unberücksichtigt. Dann ergibt sich die Liste aller Wörter, die in wenigstens einem der Dokumente vorkommen zu: (Paris Hauptstadt Frankreich Stadt Metropole). Die Dokumentvektoren haben also die Dimension 5. Für einen „echten“ Webindex hingegen wird die Dimension in der Größenordnung von mehreren Millionen liegen.

Um das Beispiel einfach zu halten, wird anstatt TF-IDF oder WDF-IDF eine andere Berechnung gewählt: Die Elemente des Vektors sind schlicht die Anzahl der Wortvorkommen im jeweiligen Dokument. Damit ergibt sich etwa der Dokumentvektor für das Dokument 3 aus Tabelle 3 zu D3 = (1 0 0 0 1). Die erste Komponente im Vektor steht für das Wort Paris, die zweite für Hauptstadt usw. und schließlich steht die fünfte Komponente für Metropole. Mit diesen Vektoren lässt sich nun beliebig rechnen und Texte können wie am Autobeispiel gezeigt klassifiziert und geclustert werden, es lassen sich die Ähnlichkeiten von Texten ermitteln und was das Arsenal an Machine-Learning-Verfahren noch alles hergibt.

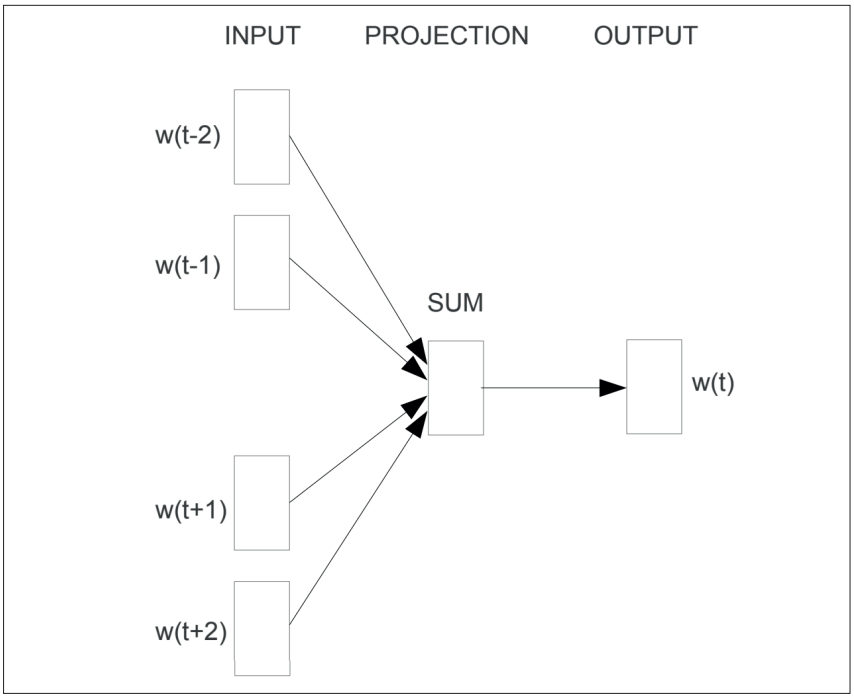

Abb. 3: Architekturmodell eines von word2vec eingesetzten neuronalen Netzwerks
(Quelle: https://arxiv.org/pdf/1301.3781.pdf)

Neuronale Netze am Vormarsch

Die bislang dargestellten Methoden sind eher konventionelle statistische Verfahren. Durch verschiedene medienwirksame Erfolge stehen heute oft auf neuronalen Netzen basierende Methoden im Vordergrund, wenn von Machine-Learning die Rede ist. Neuronale Netze zeigen verblüffende Ergebnisse in verschiedenen Bereichen: So schlägt das von Google entwickelte Programm AlphaGo inzwischen die besten Go-Spieler und die Verfahren zur Bilderkennung sind weit fortgeschritten. Für die Arbeit mit Texten kamen neuronale Netze lange Zeit kaum zum Einsatz, doch das veränderte sich spätestens mit der Freigabe der Software word2vec durch Google.

Word2vec nimmt Texte in beliebiger Sprache entgegen, analysiert die in den Texten vorhandenen Wortzusammenhänge und erstellt dann für jedes Wort, das in den Texten vorkommt, einen Vektor. Ein solches Verfahren, das für Wörter möglichst geeignete Vektoren findet, wird als Word Embedding bezeichnet. Die Dimension der Vektoren kann vom Nutzer vorgegeben werden und liegt sinnvollerweise meist irgendwo zwischen 10 und 500. Word2vec berechnet dabei die Vektoren so, dass Begriffe mit ähnlicher Bedeutung durch ähnliche Vektoren repräsentiert werden. Da nun jedes Wort durch einen Vektor vertreten wird, lässt sich mit Wörtern wie mit Vektoren rechnen, was interessante Einblicke ermöglicht.

Rechnen mit Wörtern

In der englischsprachigen Literatur zu diesem Thema findet sich als Musterbeispiel diese Rechnung: king – man + woman = queen. Übersetzt heißt das so viel wie: Zieht man von einem König alle männlichen Eigenschaften ab und nimmt die Eigenschaften einer Frau hinzu, so erhält man eine Königin. Dies ist durchaus plausibel; das Verblüffende daran ist, dass word2vec diesen Zusammenhang herstellt, ohne eine Idee zu haben, was ein König überhaupt ist.

Lässt man word2vec auf die deutsche Wikipedia los, ergeben sich ganz ähnliche Erkenntnisse. Einige besonders aussagekräftige Resultate sind in Tabelle 4 zusammengestellt. Interessant ist, dass die Königin nur als drittbestes Ergebnis auftaucht. Dies dürfte damit zusammenhängen, dass im deutschen Sprachraum Königinnen weitaus unbedeutender waren als etwa in England.

Die in der Tabelle aufgeführte Rechnung Canberra – Australien + Kanada eignet sich dazu, Hauptstädte beliebiger Länder zu ermitteln. Wird in der Formel Kanada durch Ghana ersetzt, liefert word2vec Accra als Hauptstadt Ghanas. Im Versuch mit zehn zufällig gewählten Ländern konnte das System siebenmal die richtige Hauptstadt finden.

Wieso funktioniert das? Word2vec berechnet die Wortvektoren so, dass möglichst viel Sinn aus den zugrundeliegenden Texten erhalten bleibt. Damit sind nicht nur bedeutungsähnliche Wörter nahe zusammen platziert, sondern auch die Vektoren, die von einem Wort zu einem anderen weisen, haben eine gewisse Bedeutung. So ergibt offenbar der Ausdruck Canberra – Australien die Bedeutung ist

Rechnung	nächstgelegene Vektoren
König – Mann + Frau	Gemahlin, Regentin, Königin
Königin – Frau + Mann	König, Leibwache, Königspaar
besser – gut + hoch	höher, niedriger, hohe
Yandex – Russland	Suchmaschine, Gmail, Yahoo
Canberra – Australien + Kanada	Ottawa, Ontario, Montreal

Tabelle 4

Hauptstadt von. In Abbildung 4 sind mehrere solcher Ist-Hauptstadt-von-Vektoren dargestellt und es ist deutlich zu erkennen, dass diese drei Vektoren sehr ähnlich sind.

Schon 150.000 Tweets reichen

Doch nicht nur mit einem riesigen Textkorpus wie der Wikipedia, die aus immerhin 5GB Textdaten allein in der deutschen Version besteht, lassen sich Erkenntnisse gewinnen. Eine Auswertung von etwa 150.000 Tweets deutscher Suchmaschinenoptimierer zeigt, dass word2vec alleine aus diesen kurzen Textschnipseln einschlägige Konferenzen, Tools, Suchmaschinen und Internetkonzerne in Gruppen zusammenfassen kann (Abbildung 5).

Nun ist diese Software mehr als nur eine interessante Spielerei. Paul Haahr, einer der führenden Entwickler der Ranking-Algorithmen von Google, erklärte 2016 auf einer Konferenz: „word2vec ist ein Teil von dem, was RankBrain macht“ („word2vec is one layer of what RankBrain is doing“), und der Entwickler von word2vec ist zugleich Hauptautor des RankBrain-Papers. Dummerweise sagte Haahr auch, dass Google selbst nicht ganz versteht, was RankBrain genau macht; entsprechend schwierig ist es für Außenstehende zu durchschauen. Trotzdem sollten die oben gezeigten Beispiele verdeutlichen, dass der Ranking-Algorithmus mithilfe von word2vec nun auch auf detailliertes Wissen zurückgreifen kann. Googles Forscher schreiben dazu: „Word2vec kann genutzt werden, automatisch Fakten zu extrahieren, aber auch um die Korrektheit existierender Fakten zu überprüfen.“

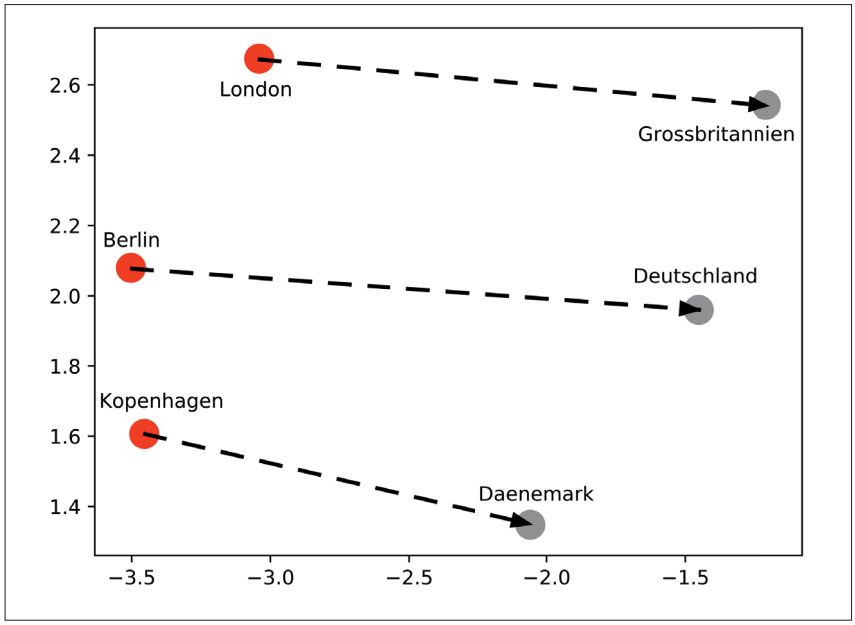

Abb. 4: Die Pfeile stellen die Beziehung X-ist-Hauptstadt-von-Y dar; es ist deutlich zu erkennen, dass sich die drei Pfeile ähneln und deshalb einen Vektor repräsentieren, der die Ist-Hauptstadt-von-Eigenschaft darstellt

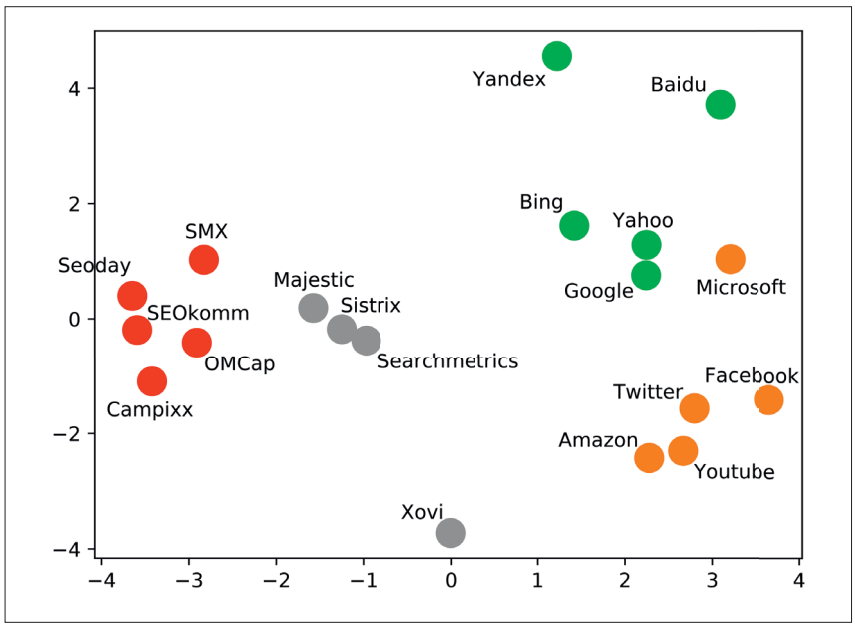

Abb. 5: Wenn deutsche SEOs twittern, geht es um alle möglichen Themen, trotzdem kann word2vec Konferenzen, Tools, Suchmaschinen und sonstige Internetkonzerne in Gruppen einordnen

Online-Marketer sollten sich der Tragweite dieser Entwicklung bewusst sein. Optimierung für einzelne Keywords ohne Rücksicht auf die Suchabsicht des Nutzers oder der Einsatz fragwürdiger Inhalte mit zweifelhaften Fakten mag heute noch weitgehend funktionieren. Doch Entwicklungen wie word2vec verschaffen dem Algorithmus ein Weltwissen, das eine Optimierung, die nicht konsequent an den Bedürfnissen der Nutzer ausgerichtet ist, zunehmend zum Scheitern verurteilt. ¶